

ATM 망에서 UPC를 이용한 트래픽 제어방법의 성능평가를 위한 시뮬레이터의 개발에 관한 연구

A Study on the Development of Simulator
for Performance Evaluation of Traffic Control
using UPC Algorithm in ATM Network

김문선**, 채기준***

Moon Sun Kim, Ki Joon Chae

Abstract

It is necessary that we should control the traffic to not only efficiently use the rich bandwidth of ATM network but also satisfy the users' various requirements for service quality. However, it is very difficult to decide which control mechanism would be applied in real network because there are various types of ATM traffic and traffic control mechanisms. In this paper, a smart simulator is developed to analyze the performance of a UPC(Usage Parameter Control) mechanism which is a typical traffic control mechanism.

The simulator consists of a user interface that supports a menu-driven input form and a simulation program that is executed with the user's input parameters. Especially, the simulator establishes more powerful and flexible simulation environment since it supports a more complex simulation applying various source traffic to several different UPC mechanisms at the same time and allows an arbitrary user-defined traffic in addition to some well-known traffic.

* 본 연구는 1996년도 한국학술진흥재단의 자유공모과제 지원으로 수행되었음.

** 한국 보쉬 기술연구소

*** 이화여자대학교 컴퓨터학과

1. 서론

광대역 종합정보 통신망(Broadband -Integrated Services Digital Network : B-ISDN)을 효율적으로 실현하기 위한 핵심 교환 기술로서 고안된 비동기 전송방식인 ATM(Asynchronous Transfer Mode)에 대한 연구가 국내외적으로 활발하게 진행되고 있다 [1]. 이는 통신하고자 하는 정보의 트래픽 특성에 무관하게 음성, 데이터, 영상 등의 트래픽을 통일적이고 효율적으로 하나의 망으로 전송하는 개념이다. 광대역 종합 정보 통신망에서의 서비스들은 광범위한 트래픽 특성과 성능, 서비스 품질을 요구하는데 ATM은 이를 위해 모든 정보를 53바이트라는 고정된 크기의 셀(Cell)로 분할하여 통계적 다중화 방법으로 전송한다[2]. 따라서 이러한 ATM 방식으로 구성된 ATM 망은 효율적인 링크의 사용을 가능하게 한다. 그러나 ATM 망이 아무리 풍부한 대역폭을 제공한다 할지라도 망자원을 효율적으로 사용하고 다양한 트래픽 특성을 갖는 호의 서로 다른 서비스 품질 요구사항을 만족시키기 위해서는 트래픽 제어가 필요하다[3][8].

ITU-T(Telecommunication Standardization Sector of International Telecommunications Union)에서는 이러한 트래픽 제어 기능으로 호 수락 제어(Call Admission Control), 망자원 관리(Resource Management), 사용 상황 감시 제어(Usage Parameter Control), 폭주 제어(Congestion Control), 우선 순위 제어(Priority Control) 등을 권고하고 있다[7]. 특히, 사용 상황 감시 제어는 호 수락시 설정된 트래픽 파라미터 값을 초과하여 셀을 전송하는 경우 망 내에서 폭주 상태를 초래할 가능성이 있으므로 반드시 필요하다. 따라서 폭주 상태를 미연에 방지하기 위한 예방적 차원에서 ATM 망으로 들어오는 트래픽의 특성을 분석할 필요가 있으며, 다양한 호의 트래픽에 대하여 여러 가지 사용 상황 감시 제어 방법을 비교 분석하여 최적의 제어 방법을 찾아내는 것이 요구된다.

이를 위하여 기존 실험에서는 필요할 때마다 특정한 UPC 알고리즘과 소스 트래픽을 모델링하여 시뮬레이션을 수행하곤 했는데, 이러한 작업은 주어진

소스 트래픽에 가장 효과적인 UPC 방법을 찾아내기 위하여 다양한 시뮬레이션을 반복적으로 수행하는 경우에 있어서 매우 비효율적일 수 있다. 따라서 본 논문에서는 다양한 트래픽에 대하여 여러 가지 사용 상황 감시 제어 방법들을 효율적이고 편리하게 모델링하며 그에 대한 시뮬레이션을 제공함으로써 망 효율의 성능을 비교 분석하기에 유용한 시뮬레이터를 개발하고자 한다.

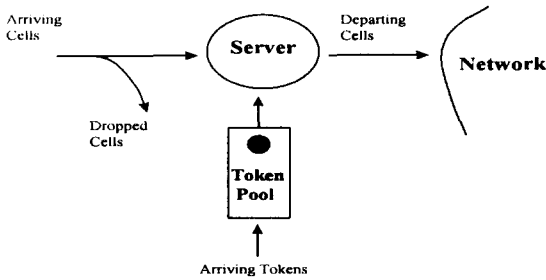
본 논문의 구성은 다음과 같다. 제1절의 개괄적인 서론에 이어 제2절에서 논문의 구현 대상인 UPC 방법에 관하여 보다 자세히 살펴보고, 제3절에서 실제 시뮬레이터의 구현 과정 및 실행 과정을 설명한 뒤, 마지막 제4절에서 결론 및 향후 연구 과제를 제시하기로 한다.

2. UPC 알고리즘

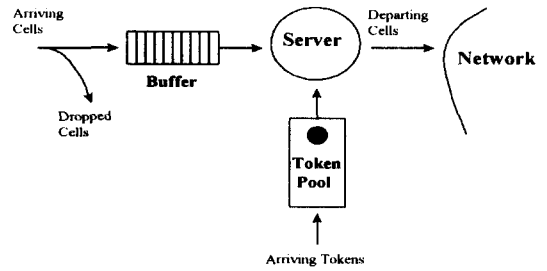
2.1 리키버킷 방식

리키버킷 알고리즘은 기본적으로 리키율과 토큰 풀 크기라는 두 가지 파라미터로 정의된다. 이 방식은 토큰 풀로부터 토큰을 받은 셀만을 망으로 전송하고, 토큰 풀이 비었을 때 도착한 셀에 대해서는 적절한 조치를 취함으로써 사용자 트래픽이 미리 신고된 트래픽 규약을 준수하도록 제어한다. 토큰 풀이 비었을 때 도착한 셀은 미리 신고한 트래픽 특성을 위반한 셀로 간주되어 <그림 1>과 같이 무조건 폐기되거나 또는 <그림 2>와 같이 태그(tag)를 달고 망으로 전송되어 폭주 발생 시에 우선적으로 폐기되도록 구현할 수 있다.

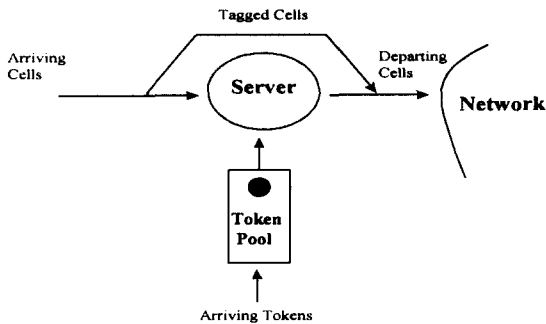
이처럼 셀의 전송 여부를 결정하는 토큰은 일정한 속도(리키율)로 생성되어 토큰 풀에 차곡차곡 쌓이며, 토큰 풀이 꽉 찼을 때 생성된 토큰은 그대로 폐기된다. 이때 토큰이 생성되는 속도인 리키율은 망의 평균 허용 속도를 기준으로 정하지만 보통 이 값보다 크게 정해야 효율성이 있으며, 토큰 풀 크기는 허용 가능한 최대 버스트 길이를 기준으로 정한다.



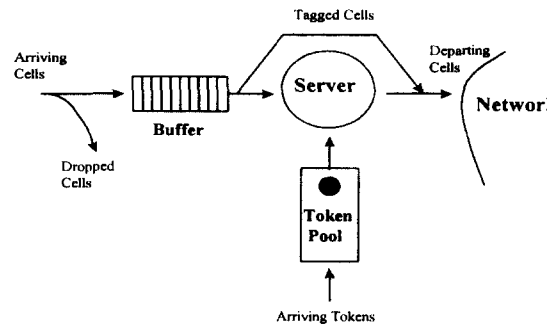
<그림 1> 리키버킷 알고리즘 (단순 폐기형)



<그림 3> 리키버킷 알고리즘 (버퍼 폐기형)



<그림 2> 리키버킷 알고리즘 (단순 표기형)



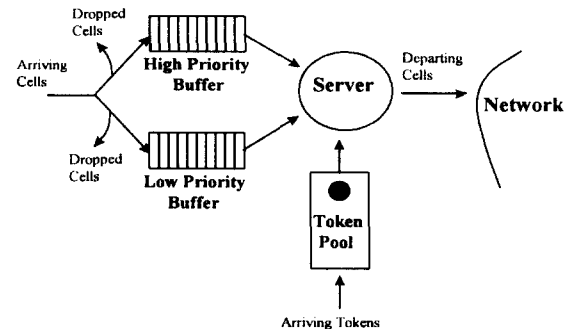
<그림 4> 리키버킷 알고리즘 (버퍼 표기형)

이러한 리키버킷 알고리즘은 1986년 Turner에 의해 처음으로 소개된 이후 다양한 방식의 변형 알고리즘들이 제안되었다[1].

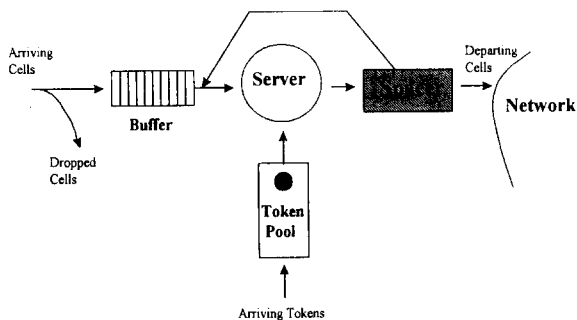
<그림 3>은 가장 대표적인 변형 알고리즘으로서, 셀이 도착했을 때 토큰 풀에 토큰이 없더라도 바로 폐기되지 않고 다음 번 토큰이 생성될 때까지 버퍼에서 기다릴 수 있도록 입력 버퍼를 첨가한 것이다. 이 방식의 경우, 기본적인 리키버킷 알고리즘에 비해 폐기되는 셀의 수가 줄어들기 때문에 전체 셀 손실율을 감소시킬 수 있지만 버퍼에 의한 비용이 증가하고 버퍼 지연으로 인한 셀 지연을 가중시킨다는 단점이 있다. 이때 셀을 폐기시키는 것 대신에 태그를 달아 전송시킴으로써 <그림 4>와 같이 변형된 구조가 가능하다.

그리고 여러 가지 다양한 종류의 트래픽을 모두 수용하는 ATM 망의 특성 상 사용자의 서비스 품질 요구사항을 충족시키기 위해서는 트래픽의 중요도에 근거하여 셀마다 우선순위를 부여할 수 있다.

따라서 <그림 5>와 같이 입력 버퍼에 우선순위를 두고 동일한 우선순위의 셀들끼리 모았다가 처리하는 방식도 고려할 수 있다.



<그림 5> 리키버킷 알고리즘 (우선순위 버퍼형)



<그림 6> 리키버킷 알고리즘 (스페이스형)

또한 버퍼를 갖는 리키버킷 알고리즘에 <그림 6>과 같이 스페이서를 첨가하여 스페이서에 의해 제어되는 구조도 제안되었다[15]. 광대역 종합 정보 통신망은 다양한 서비스를 제공하므로 셀의 도착 간격이 트래픽마다 일정하지 않고 트래픽의 버스트 특성에 따라 제각기 다르다. 특히 버스티한 트래픽의 경우, CDV(Cell Delay Variation)가 매우 커지게 되어 그대로 통과시키면 망에서 폭주를 발생시킬 위험이 있다. 따라서 일정한 시간 간격으로 셀을 통과시키는 스페이서를 첨가함으로써 망 내에 유입되는 셀 간격을 일정하게 조절할 수 있다.

2.2 윈도우 방식

2.2.1 점핑 윈도우

먼저 점핑 윈도우 알고리즘은 고정된 시간 간격(윈도우) T 내에 도착하는 셀의 개수를 제한하는 방법으로 각각의 윈도우 T당 최대 N개의 셀이 허용되고 이전의 윈도우가 끝남과 동시에 새로운 윈도우가 시작된다. 계수가 값은 각 윈도우가 시작되는 시점에서 0으로 리셋되고, 만약 계수가 값이 N과 같아지면 그 이후로 도착하는 셀들은 다음번 윈도우가 시작될 때까지 모두 폐기되거나 CLP(Cell Loss Priority) 비트가 1로 셋트된다. 이때 윈도우의 크기가 너무 커지면 트래픽 폭주의 발생을 인식하는 데 걸리는 시간이 길어질 수 있고, 반면에 너무 작아지면 트래픽 규약을 제대로 지키고 있는 다른 사용자들에 대한 투명성(transparency)이 제공되지 않을 수도 있다. 따라서 초기에 윈도우의 크기를 적당하

게 설정하는 것이 알고리즘의 성능을 결정하는 데 있어 중요하다[1].

2.2.2 트리거 점핑 윈도우

점핑 윈도우의 변형으로 볼 수 있는 트리거 점핑 윈도우 알고리즘은 각 윈도우의 시작을 셀이 도착하는 시간과 동기화시킨다는 점만 제외하고 그 외의 동작은 앞서 소개된 점핑 윈도우 알고리즘과 동일하다.

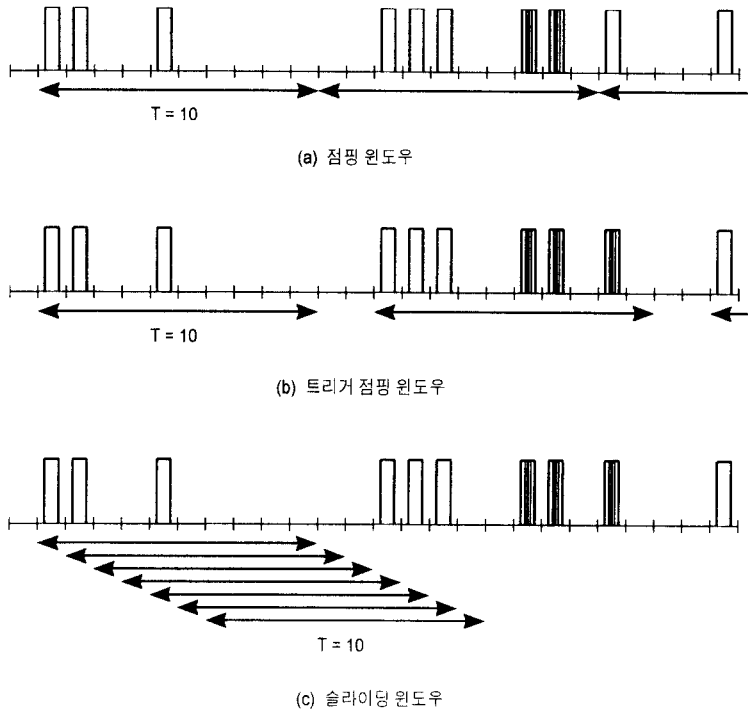
2.2.3 슬라이딩 윈도우

슬라이딩 윈도우 알고리즘은 셀 발생 시 카운터를 증가시키고 현재를 기준으로 T 이전에 발생했던 셀이 나타나면 감소시켜 지난 T 시간 동안 얼마나 많은 셀이 통과되었는지를 카운터 값을 통해 알 수 있다. 이 방법은 T 시간 동안에 발생하는 셀의 개수와 각 셀의 도착 시간을 기억하고 있어야 하기 때문에 하드웨어 구현이 복잡하다는 단점이 있다.

다음의 <그림 7>은 윈도우의 크기(T)가 10이고 각 윈도우에서 최대한으로 허용되는 셀의 개수(N)가 3일 때, 위의 세 가지 윈도우 알고리즘을 적용시킨 예제를 보여준다. <그림 7>에서 내부가 어둡게 칠해진 직사각형은 트래픽 규약을 위반하여 폐기되는 셀들을 나타낸다.

2.2.4 EWMA

EWMA 알고리즘에서 윈도우의 진행 방식은 점핑 윈도우에서의와 같다. 즉, 윈도우 크기 T는 고정적이고 새로운 윈도우는 이전 윈도우에 연이어 곧바로 시작된다. 그러나 앞서 소개된 다른 윈도우 방식들과는 달리, i번째 윈도우에서 허용되는 최대 셀의 개수 m_i 가 이전 윈도우에서 허용되었던 셀의 개수 (X_{i-1}) 와 평균 셀의 개수(m)에 대한 지수적 가중치 합계(exponentially weighted sum)에 의해 결정되기 때문에 각 윈도우마다 허용되는 최대 셀의 개수 값이 다양하다는 특징이 있다[1].



<그림 7> 윈도우 방식의 UPC 알고리즘

3. 시뮬레이터 구현

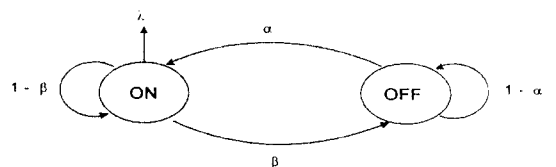
본 논문에서 구현한 시뮬레이터는 크게 시뮬레이션 프로그램 부분과 사용자 인터페이스 부분으로 구분된다. 시뮬레이션 프로그램 부분은 다시 소스 트래픽의 모델링 단계와 UPC 알고리즘의 적용 단계로 나누어지며, 이를 C 언어를 이용하여 구현하였다. 그리고 사용자 편의를 도모하기 위하여 제공되는 풀다운 메뉴 방식의 통합 환경은 UNIX 상에서 X-윈도우 프로그램을 이용하여 구현하였다. 이에 본 장에서는 시뮬레이터의 구현 내용과 함께 시뮬레이터의 동작 과정을 함께 기술하고자 한다.

3.1 소스 트래픽 모델링

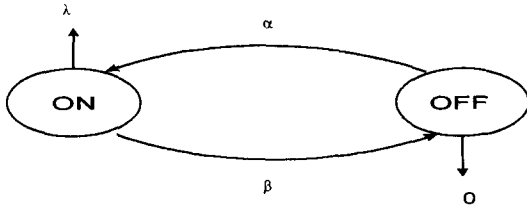
3.1.1 IBP(Interrupted Bernoulli Process) 모델

IBP 모델은 ON/OFF 모델을 기본으로 하며 <그림 8>과 같이 표현된다. 이산 모형(Discrete Model)

인 IBP 모델의 주요 특징은 연속적인 시간(time)의 흐름이 일정 길이의 타임 슬롯(time slot)들로 분할되고 각 타임 슬롯이 ON이나 OFF 상태를 갖는다는 점이다[1]. ON에서 OFF로 옮겨갈 확률이 β 이고 OFF에서 ON으로 전이할 확률이 α 라면, 평균적으로 ON 상태에 머무를 시간은 $1/(1-\beta)$ 이고 이 기간 동안에 생성되는 셀들은 평균 λ 의 베르누이 분포를 따른다.



<그림 8> IBP 모델



<그림 9> IPP 모델

3.1.2 IPP(Interrupted Poisson Process) 모델

IBP 모델의 연속 시간 버전인 IPP 모델[17]은 <그림 9>와 같이 표현된다. IPP 모델에서 OFF와 ON 상태에 머무는 시간은 각각 독립적인 지수 분포를 따른다. 따라서 <그림 9>에서 α 와 β 를 각각의 전이율로 해석할 경우 OFF와 ON 상태에 머무는 시간은 α 와 β 의 지수 분포를 따른다. 그리고 IPP 모델 역시 ON/OFF 모델을 기본으로 하기 때문에 ON 기간 동안에 평균 λ 로 셀을 생성하지만 OFF 구간에서는 셀을 전혀 생성하지 않는다.

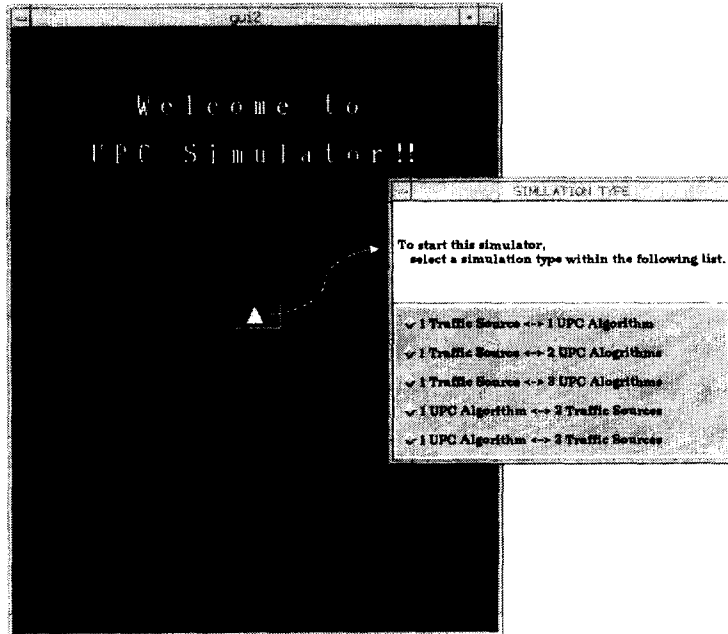
3.2 사용자 인터페이스

3.2.1 시뮬레이션 유형 선택

<그림 10>과 같이 초기 화면에서 중심부의 화살표 버튼을 클릭하면 조그만 윈도우가 뜨게 되는데, 이 윈도우는 시뮬레이터를 구동시키기 전에 사용자가 원하는 시뮬레이션 유형을 선택하도록 한다. 이는 기존에 개발된 대부분의 시뮬레이터들이 어떤 특정한 소스 트래픽과 UPC 알고리즘만을 대상으로 한다는 제한성을 극복하기 위한 것이다.

다음은 본 시뮬레이터에서 제공하는 시뮬레이션의 유형들로서, 이는 기존에 유사한 시뮬레이션을 반복적으로 수행하던 번거러움을 덜어줄 수 있다.

- ① 한 개의 소스 트래픽을 한 개의 UPC 알고리즘에 적용
- ② 한 개의 소스 트래픽을 두 개의 UPC 알고리즘에 적용
- ③ 한 개의 소스 트래픽을 세 개의 UPC 알고리즘에 적용
- ④ 한 개의 UPC 알고리즘에 두 개의 소스 트



<그림 10> 시뮬레이션의 유형 선택

래픽을 적용

⑤ 한 개의 UPC 알고리즘에 세 개의 소스 트래픽을 적용

특히, ②와 ③의 유형은 어느 특정한 소스 트래픽을 제어하는 방법으로 어떤 UPC 알고리즘이 가장 적절한지를 찾아내기 위하여 여러 개의 UPC 알고리즘을 동시에 적용시키고 그들의 제어 성능을 비교 분석하기에 용이하며, ④와 ⑤의 유형은 어느 특정한 UPC 알고리즘이 작동하고 있는 ATM 망으로 다양한 종류의 소스 트래픽들이 전송될 경우 각 트래픽의 특성에 따라 현재 적용되고 있는 UPC 알고리즘의 성능이 어떻게 달라지는지를 실험해 보기에 유용하다.

3.2.2 시뮬레이터 구동

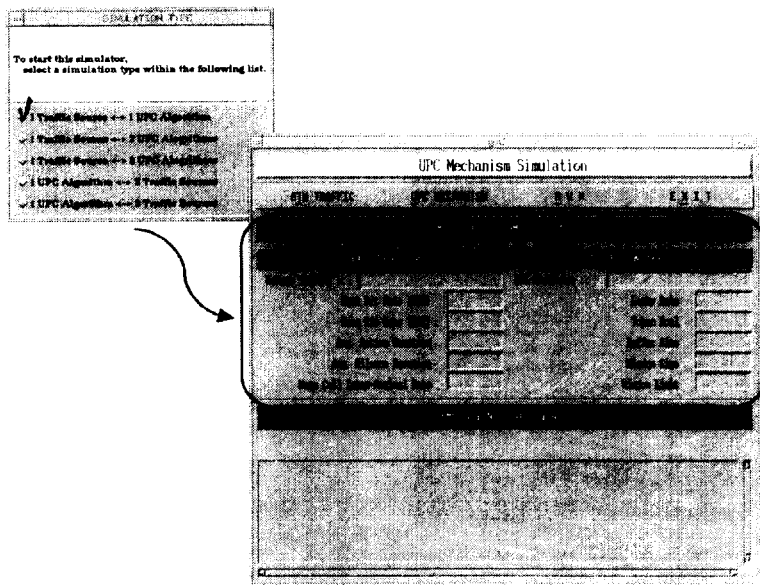
사용자가 위의 다섯 가지 유형 중에서 한 가지를 선택하면 각 경우에 해당하는 다양한 시뮬레이터 화면이 나타나게 된다. <그림 11>은 가장 기본적인 첫 번째 유형을 선택했을 경우로 트래픽 파라미터 윈도우와 UPC 윈도우가 양쪽으로 나누어진다.

그러나 사용자가 선택한 시뮬레이션 유형에 상관

없이, 전체 화면은 크게 두 개의 윈도우로 나누어지고 화면 최상단부에 풀다운 식의 메뉴 바(bar)가 위치하게 된다. 상단 윈도우는 사용자가 선택한 트래픽에 관한 여러 가지 트래픽 파라미터 값들과 UPC 알고리즘에 대하여 사용자가 직접 입력한 파라미터 값들을 보여주기 위하여 사용된다. 그리고 하단 윈도우에서는 사용자의 입력 파라미터 값들을 사용하여 주어진 시간 동안 수행한 시뮬레이션의 최종 결과를 보여주는데, 이때 성능 분석치로 셀 손실율과 전송 지연 등이 사용된다. 만일 사용자가 선택한 시뮬레이션 유형이 다중 모드일 경우(②~⑤)에는 각 경우에 대한 시뮬레이션을 별도로 수행한 뒤에 그 결과를 모아서 한번에 보여주게 된다.

3.2.3 소스 트래픽 선택

본 시뮬레이터는 사용자의 입력에 의한 소스 트래픽 이외에, 기본적으로 ATM 망에서 자주 사용되는 몇 가지 상용 서비스 트래픽들을 지원하는데, 구체적인 파라미터 값들은 다음의 표 1을 참조하기로 한다.

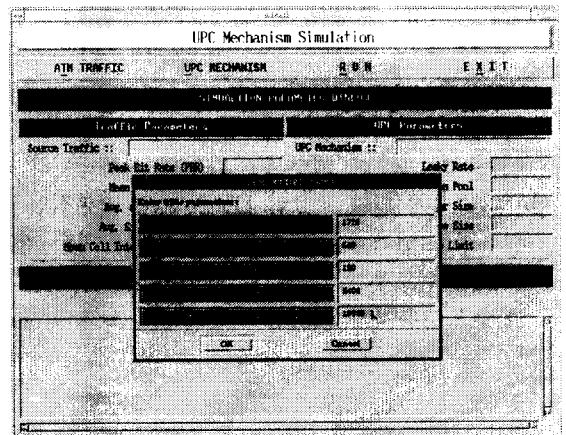


<그림 11> 시뮬레이션 유형(1) 선택

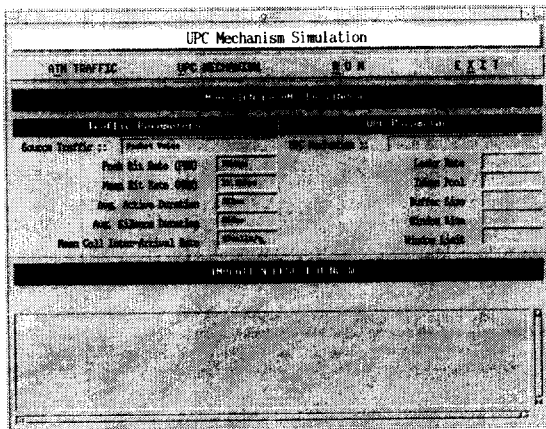
<표 1> 소스 트래픽의 종류와 특성

트래픽 파라미터	Packet voice	Still Picture	High Speed Data	Telephony
PBR	32Kbps	2Mbps	10Mbps	64Kbps
MBR	11.2Kbps	87Kbps	1Mbps	-
Avg. Burst Duration	352ms	500ms	13ms	960ms
Avg. Silence Duration	650ms	11000ms	117ms	1690ms
Mean Cell Inter-Arrival Rate	83.3cells/s	5208.4cells/s	26041.7cells/s	166.7cells/s

사용자가 위의 상용 소스 트래픽들 중에서 한 가지를 소스 트래픽으로 선택한다면, 각 트래픽에 대하여 이미 주어져 있는 최대 비트 전송율(Peak Bit Rate : PBR), 평균 비트 전송율(Mean Bit Rate : MBR), 평균 활성화 구간 길이(Average Active Duration), 평균 침묵 구간 길이(Average Silence Duration), 평균 셀 간 도착율(Mean Cell Inter-Arrival Rate) 등의 파라미터 값들이 자동적으로 윈도우 안에 표시된다. 반면에 사용자 정의 트래픽을 선택한 경우라면 <그림 13>와 같이 사용자가 직접 트래픽 파라미터 값들을 입력하여 채우게 된다.



<그림 13> 사용자 정의 트래픽



<그림 12> 소스 트래픽 선택 (단일 모드)

3.2.4 UPC 알고리즘 선택

UPC 알고리즘을 선택하고 그에 필요한 파라미터 값들을 입력한다. 이때 사용자가 입력하는 파라미터의 종류는 UPC 알고리즘에 따라 조금씩 다르다. 기본형 리키버킷의 경우에는 리키율과 토큰 풀 크기만을 입력하면 되지만 버퍼를 갖는 리키버킷의 경우에는 이외에 버퍼 크기를 추가적으로 입력해야 하고, 윈도우 기반의 알고리즘 경우에는 윈도우 크기와 각 윈도우에서 허용되는 셀의 최대 개수에 관한 값을 입력해야 한다.

<표 2> UPC 기법과 파라미터

UPC 알고리즘		입력 파라미터
리키버킷 방식	기본형	리키울, 토큰풀 크기
	버퍼형	리키울, 토큰풀 크기, 버퍼 크기
	우선순위 버퍼형	리키울, 토큰풀 크기, 버퍼 크기
윈도우 방식	점핑 윈도우	윈도우 크기, 허용 셀 개수
	트리거 점핑 윈도우	윈도우 크기, 허용 셀 개수

3.2.5 시뮬레이션 시간 입력

소스 트래픽과 UPC 알고리즘이 결정되고 그에 필요한 파라미터 값들의 입력이 완료된 후에야 비로서 시뮬레이션을 진행시키게 되는데, 이때 시뮬레이션의 수행 기간을 사용자가 지정해줄 수 있기 때문에 보다 다양한 시뮬레이션이 가능해진다.

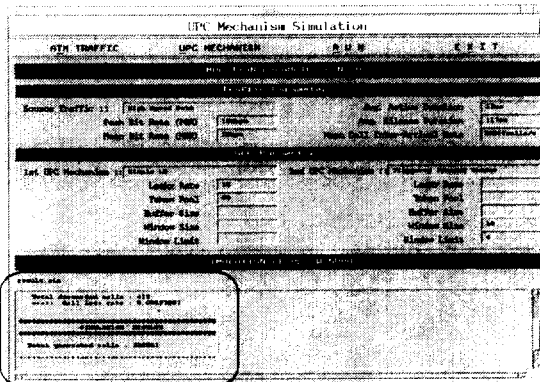
3.2.6 시뮬레이션 결과 출력

다음의 <그림 14>는 시뮬레이션 유형 ②의 예로서 High Speed Data 소스를 기본형 리키버킷과 트리거 점핑 윈도우 방법에 적용시킨 경우이다. 즉, 하나의 소스 트래픽을 서로 다른 두 가지의 UPC 알고리즘에 동시에 적용시킴으로써 각 알고리즘의 성능을 비교 분석하기 위한 것이다. 성능 비교를 위한 시뮬레이션 결과 자료로는 셀 손실율과 전송 지연시간을 보여준다. (<그림 14>는 트리거 점핑 윈도우 방

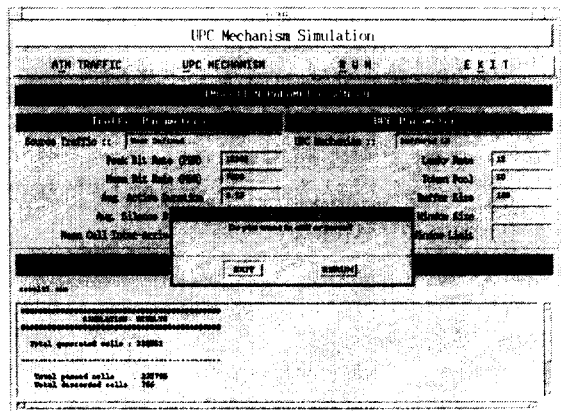
법의 셀 손실율을 보여주나 화살표를 이동하여 다른 성능 자료를 볼 수 있음) 즉, 본 시뮬레이터는 시뮬레이션 시간 동안에 생성된 셀의 총 개수와 각 알고리즘에 따라 폐기되어진 셀의 개수를 하나의 윈도우 안에서 별도로 표시함과 동시에 각각의 셀 손실율을 계산해 주며 전달되어진 셀들의 전송 지연시간을 보여 줌으로써 서로 다른 알고리즘의 성능을 비교하고 분석하는 데에 필요한 데이터를 제공한다.

3.2.7 시뮬레이터 종료

시뮬레이터의 종료를 확인하는 메시지가 출력된다. <그림 15>에서 보는 바와 같이 메시지와 함께 종료 버튼과 재실행 버튼이 나타난다.



<그림 14> 시뮬레이션 결과 출력



<그림 15> 시뮬레이터 종료

이때 사용자가 시뮬레이터의 종료를 원한다면 왼쪽 하단부의 종료 버튼을 선택하면 되고, 만약에 또 다른 시뮬레이션을 계속해서 수행하고자 한다면 오른쪽의 재실행 버튼을 선택하면 된다. 재실행 버튼이 선택되면 이전에 수행되었던 시뮬레이션 내용은 모두 지워지고 사용자는 새로운 시뮬레이션을 위하여 새로운 입력을 할 수 있게 된다.

4. 결론

본 연구를 통해 개발된 시뮬레이터는 여러 종류의 트래픽을 다양한 UPC 방법으로 전송할 경우 각각의 성능을 비교 분석함으로써 주어진 상황에 가장 적합한 방법을 선택하기 위한 하나의 도구로 사용될 수 있다. 물론 이전에도 관련 연구 분야에서 특정한 UPC 방법의 성능 분석을 위하여 시뮬레이션이 수행되기도 했지만 기존의 실험들은 대부분이 일회적인 단일 프로그램 수준에 그쳤다는 한계점을 가지고 있다. 반면에, 본 시뮬레이터는 사용자의 다양한 요구를 한번에 수용할 수 있도록 통합적인 시뮬레이션 환경을 제공하고 있으며, 무엇보다도 사용자 편의의 인터페이스를 갖추고 있기 때문에 시뮬레이션으로 인한 사용자의 부담을 덜어줄 수 있다.

그리고 더 나아가, 소스 트래픽 모델링 기법의 종류를 좀더 다양하게 늘리고 그 내용을 하나의 독립적인 메뉴로 구성하여 사용자가 직접 선택할 수 있도록 개발한다면 사용자의 만족도 뿐만 아니라 시뮬레이터의 성능까지도 크게 향상시킬 수 있을 것이다. 또한 시뮬레이션의 결과가 단순한 텍스트 형태로 출력되는 점을 개선하여 UNIX 상에서 그래프로 출력할 수 있는 방안을 개발한다면, 본 시뮬레이터의 최종 목표인 성능 분석 기능을 한층 더 강화시킬 수 있으리라 기대한다.

참고문헌

- [1] R. O. Onvural, "Asynchronous Transfer Mode Networks : Performance Issues," 2nd edition, Artech House, Boston.London, 1994.
- [2] R. Handel, M. N. Huber, S. Schroder, "ATM Networks : Concepts, Protocols, Applications," 2nd edition, Addison-Wesley, 1994.
- [3] J. J. Bae, T. Suda, "Survey of Traffic Control Protocols in ATM Networks," GLOBECOM'91, pp.300.1.1-300.1.6, 1991.
- [4] T. Okada, H. Ohnishi, N.Morita, "Traffic Control in Asynchronous Transfer Mode," IEEE Comm. Mag., pp.58-62, Sep. 1991.
- [5] N. Yamanaka, Y. Sato, K. Sato, "Performance Limitation of Leaky Bucket Algorithm for Usage Parameter Control and Bandwidth Allocation Methods," IEICE Trans. on Com., Vol.E75-B, No.2, pp.82-86, Feb. 1992.
- [6] A. V. Vasilakos, A. F. Atlasis, "LB-SELA:Rate-Based Access Protocol for ATM Networks," IEEE INFOCOM'94, pp.5c.2.1-5c.2.8, 1994.
- [7] ITU-T Recommendation I.371, "Traffic Control and Congestion Control in B-ISDN," Mar. 1994.
- [8] W. Wang, T. N. Sadawi, K. Aihara, "Bandwidth Allocation for ATM Networks," IEEE ICC'90, pp.439-442, 1990.
- [9] E. Rathgeb, "Modelling and Performance Comparison of Policing Mechanisms for ATM Networks," IEEE JSAC., Vol.9, No.3, pp.325-334, Apr. 1991.
- [10] M. Butto, E. Cavallero, A. Tonietti, "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks," IEEE JSAC., Vol.9, No.3, pp.335-342, Apr. 1991.
- [11] M. Ilyas, H. T. Mouftah, "Performance Evaluation of Congestion Avoidance in

- Broadband ISDNs," IEEE ICC'90, pp.727-731, 1990.
- [12] T. Okuda, H. Akimaru, K. Nagai, "Performance Evaluation for multiclass Traffic in ATM Systems," ICC, pp.847-851, 1990.
- [13] H. Heltes, D. H. Lucantoni, "Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Stochastic Multiplexor Performance," IEEE JSAC., Vol.6, No.4, pp.856-863, 1986.
- [14] R. Handel, "Evolution of ISDN towards Broadband ISDN," IEEE Network Mag., pp.7-13, Jan. 1989.
- [15] D. Hong, T. Suda, "Congestion Control and Prevention in ATM Networks," IEEE Network Mag., pp.10-16, Jul. 1991.
- [16] A. S. Acampora, "An Introduction to Broadband Networks," Plenum, 1994.
- [17] H. Yamada, S. Sumita, "A Traffic Measurement Method and its Application for Cell Loss Probability Estimation in ATM Networks," IEEE JSAC., Vol.9, No.3, pp.315-324, Apr. 1991.
- [18] T. Okada, H. Ohnishi, N. Morita, "Traffic Control in Asynchronous Transfer Mode," IEEE Comm. Mag., pp.58-62, Sep. 1991.
- [19] F. Guillemin, P. Boyer, L. Romoef, "The Spacer-Controller: Architecture and First Assessments," IFIP Workshop, pp.294-304, 1992.
- [20] A. W. Berger, "Performance Analysis of a Rate Control Throttle Where Tokens and Jobs Queue," IEEE INFOCOM'90, pp.30-38, 1990.

● 저자소개 ●



김문선

1996년 이화여자대학교 컴퓨터학과 학사
1998년 이화여자대학교 컴퓨터학과 석사
1998년~현재 한국 보쉬 기술연구소 연구원
관심 분야 : 고속통신망, 성능평가



채기준

1982년 연세대학교 수학과 학사
1984년 미국 Syracuse Univ. 전자계산학과 석사
1990년 미국 North Carolina State Univ. 컴퓨터공학과 박사
1990~1992년 미국 해군사관학교 전자계산학과 조교수
1992년~현재 이화여자대학교 컴퓨터학과 부교수
관심분야 : 고속통신망, LAN, 무선통신망, 망관리, 성능평가, 정보보호시스템